TAM VALLEY DEPOT

Quad-LN User Manual

V1.03

N3IX Engineering
15 Oct 2012

Contents

1 Basics 1
1.1 Overview 1
1.2 Servos 1
1.3 External Power Supply 1
1.4 Onboard LED Indicators 1
1.4.1 SERVO 1 – SERVO 4 LEDs 1
1.4.2 LN POWER LED 2
1.4.3 LN MESSAGE LED 2
1.4.4 LN ACTIVE LED 2
1.5 Frog Power 2
1.6 Inputs 2
1.7 Addresses 2
1.7.1 Device Address 2
1.7.2 Servo Start Address 2
1.7.3 Sensor Start Address 3
1.8 Groups 3
1.8.1 Turnout Position Feedback 3
1.8.2 Panel Indicators 3
2 Quick Start 4
2.1 Connecting Power 4
2.2 Setup without JMRI 4
2.2.1 Servo Start Address 4
2.2.2 Manual Alignment of Travel, Speed and Orientation 4
2.2.3 Automatic Alignment of Travel 5
2.2.4 Orientation 6
2.2.5 Device Address 6
2.3 Setup with JMRI 6
2.3.1 Create roster entry and set the Device Address 7
2.3.2 Servo Start Address and Sensor Start Address 7
2.3.3 Travel and Speed 8
2.4 Factory Reset 8
2.4.1 Restore Selective Parameters to Factory Defaults using JMRI 8
2.4.2 Restore All Parameters to Factory Defaults using the Remote Align Board 8
2.4.3 Restore Selective Parameters to Factory Defaults without using JMRI 9
3 Servo Control 9
3.1 Travel 9

3.1.1 Closed Position 9
3.1.2 Thrown Position 9
3.1.3 Orientation 9
3.1.4 Speed 10
3.1.5 RapidStart 10
3.1.6 Directional Speed 10
3.1.7 Thrown Speed 11
3.1.8 Thrown RapidStart 11
3.2 Lock 11
3.3 Midpoint 12
3.4 Message - Servo Position Feedback without sensors 13
3.4.1 Exact feedback (sim) 13
3.4.2 Output level feedback 13
3.5 Cascade 13
3.5.1 Trigger 14
3.5.2 Action 14
3.5.3 Turnout 14
3.5.4 Cascade Examples 14
4 Main and Aux IO lines 15
4.1 Input Response 15
4.1.1 Trigger 15
4.1.2 DCC Freeze 16
4.1.3 Message 16
4.2 Indication 16
4.2.1 LED Mode 16
4.2.2 LED Sense 17
4.3 Action 17
4.4 Secondary Message 17
4.4.1 Type 18
4.4.2 Device 18
4.4.3 Condition 18
4.4.4 Number 18
4.4.5 Secondary Message Examples 18
5 Routes 19
5.1 Type 19
5.2 Entries 19
5.3 Route Examples 19
6 The Quad-LN Pane 20

6.1 Addresses 20
6.1.1 Long Address 20
6.1.2 Desired Servo Start Address 20
6.1.3 Desired Sensor Start Address 20
6.2 Interrogate Options 21
6.2.1 Input Interrogate 21
6.2.2 GPON Interrogate 21
6.3 Servo State Memory 21
6.4 Retry on DCC Busy 21
6.5 Send Output Off Switch Request 22
6.6 Local Action Visibility 23
6.7 AutoAlign Current 23
6.8 Lock Indicator 23
6.9 Device Information 23
6.9.1 Manufacturer ID 23
6.9.2 Product ID 23
6.9.3 Hardware Version 24
6.9.4 Software Version 24
6.9.5 Build Number 24
6.10 Other info for JMRI use only 24
7 Updating the Firmware 24
8 CV Tables 26
8.1 Full CV Table 26
8.2 Long Address CV Format 32
8.3 Configuration CV Format 32
8.4 Servo Speed CV Format 32
8.5 Servo Output Option CV Format 33
8.6 Servo Cascade Address CV Format 33
8.7 Main IO Option CV Format 34
8.8 Main IO Action CV Format 34
8.9 Main IO Secondary CV Format 34
8.10 Aux IO Option CV Format 35
8.11 Aux IO Action CV Format 35
8.12 Aux IO Secondary CV Format 35
8.13 Route CV Format 36
9 JMRI and Turnout Feedback 37
9.1 Switch Request 37
9.2 Turnout feedback 37
9.2.1 Approach 1. OUTPUT form of OPC_SW_REP 37

9.2.2 Approach 2. INPUT form of OPC_SW_REP 38
9.2.3 Comments 39

1 Basics

1.1 Overview

The **Quad-LN** is a stationary decoder that can drive 4 servos for controlling turnouts. The servos can be actuated using pushbutton or toggle switch inputs. A single input can control one servo or multiple servos. The **Quad-LN** can also drive 4 remote relay boards.

The **Quad-LN** can display the servo state on a Tam Valley Depot Fascia Controller or drive user control panel LEDs. Besides showing whether a turnout is Closed or Thrown, the **Quad-LN** can indicate when a servo is in motion (without requiring Turnout Position Feedback sensors!) and when it is Locked.

The **Quad-LN** is a Loconet device. Servos can be controlled via Loconet Switch commands. Turnout position can be reported to software such as JMRI. The **Quad-LN** even provides local control Locking capability so that local pushbutton or toggle switch servo control can be disabled on an individual servo basis whenever desired by just sending a Loconet Switch command.

Quad-LN inputs can be wired to turnout position sensors and then generate Loconet Turnout Feedback messages. Inputs can also be wired to detectors and then report block occupancy using Loconet General Sensor messages. Almost any device with a contact closure or logic-level signal output can be wired to an available **Quad-LN** input and then reported via Loconet.

The **Quad-LN** also includes Cascade and programmable Routes/ Selectors features that take advantage of the Loconet to implement layout control functions such as yard ladder control and track selection. Although setting up these features is more complex that just configuring a single turnout, using the **Quad-LN** template in JMRI greatly simplifies things.

1.2 Servos

The **Quad-LN** drives up to 4 R/C airplane servos for controlling turnouts or performing other layout animation tasks. Independent control over the travel and speed for each servo, plus a RapidStart feature, make turnout movement more realistic.

1.3 External Power Supply

The Quad-LN requires an external DC power supply between +5V and +24V. Do not exceed +24V DC. See section 2.1 for connection details.

1.4 Onboard LED Indicators

There are 7 LEDs on the **Quad-LN**.

1.4.1 SERVO 1 – SERVO 4 LEDs

The SERVO LEDs indicate the state of the relay output of each servo. Each LED is turned on when its Servo is CLOSED and turned off when its Servo is THROWN. When a servo is in motion, the LED changes state when the servo reaches its travel midpoint.

1.4.2 LN POWER LED

The LN POWER LED indicates that Loconet Rail Sync is present and the opto-isolated Loconet interface is powered. This LED should be ON when a Loconet cable is plugged into the **Quad-LN**.

1.4.3 LN MESSAGE LED

The MSG LED flashes when any Loconet message traffic is seen, other than messages sent by the **Quad-LN**. If this LED never flashes then there is probably a Loconet connection issue.

1.4.4 LN ACTIVE LED

The ACTV LED flashes when any Loconet message is sent by the **Quad-LN** or when the **Quad-LN** acts upon any incoming Loconet message. This LED also flashes when the DCC state changes.

1.5 Frog Power

The Quad-LN can drive 4 remote relays for powering turnout frogs and other uses. The relay state changes at the servo travel midpoint to reduce problems with electrical shorts during movement with certain types of turnouts.

1.6 Inputs

The Quad-LN provides 4 “Main” Inputs and 4 “Aux” Inputs. Both types of inputs are fully configurable, including when to respond and what action to take. Input actions occur only when an input changes. In addition to controlling servos, inputs can provide Turnout Feedback or General Sensor reports, send Track Power On/Off messages, or send an Emergency Stop message. When needed for advanced layout control functions, inputs changes can even produce Switch commands or arbitrary General Sensor messages.

The Quad-LN has special features that allow the input lines to also drive LED indicators or other devices. Since each “input” also has an “output” capability, each connection is called an “IO” line in this manual (IO = Input / Output).

1.7 Addresses

1.7.1 Device Address

The Quad-LN requires a Device Address so its features can be configured via Loconet **Ops Mode** programming. This address is just for device programming, and is independent from the Servo and Sensor addresses which are described in 1.7.2 and 1.7.3 below. The valid range for the Device Address is 0-16383 and it must be unique.

Typically a 5 digit extended address is used for the Device Address. This avoids conflict with locomotives and other devices when doing **Ops Mode** programming. The default Device Address is 11000. When first installing Quad-LN’s remember to connect just one unprogrammed unit at a time to Loconet and change the Device Address to a unique address.

1.7.2 Servo Start Address

The Quad-LN responds to Loconet Switch commands at addresses starting at the Servo Start Address. The Quad-LN uses 8 consecutive switch addresses, with the first 4 assigned to the Servos and the last 4

used for the optional Servo Locks. The Servo Start Address can be from 1 to 2033 in steps of 8. The value entered will be adjusted down if needed to the nearest group-of-8 boundary (1, 9, 17 … 2025, 2033).

1.7.3 Sensor Start Address

The Quad-LN reports the status of its inputs using Loconet Sensor addresses starting at the Sensor Start Address. The Quad-LN uses 8 consecutive sensor addresses, assigned in sequence to Aux IO1, Main IO1, Aux IO2, Main IO2, and so on. The Sensor Start Address can be from 1 to 4089 in steps of 8. The value entered will be adjusted down if needed to the nearest group-of-8 boundary (1, 9, 17 … 4081, 4089).

1.8 Groups

The Quad-LN has 4 functional Groups. Each Group consists of a Servo, Main IO line and Aux IO line. As shipped, the Main IO line in each Group is configured for a momentary pushbutton and changes the state of the Servo on each button press. The Aux IO line is configured for a steady input, such as a toggle switch, and commands the Servo to follow the input.

Any Input can be configured to control any Servo. In fact any input can control as many of the Quad-LN Servos as desired, or it can serve other functions and not control any Quad-LN servo. There are some Quad-LN features (LED indication, Turnout Position feedback, etc) that only work when an Input is used with the Servo in the same Group, so it is best to start with that arrangement when learning about the Quad-LN.

1.8.1 Turnout Position Feedback

When using Standard (Indirect) Turnout Position Feedback, wire the Closed sensor to the Main IO line in the same group as the Servo. When using Exact Turnout Position Feedback, wire the closed sensor to the Main IO line and wire the Thrown sensor to the Aux IO line in the same group as the Servo.

Note: The Quad-LN can provide Turnout Position Feedback without any sensors. See 0 below for detailed info.

1.8.2 Panel Indicators

The Quad-LN can drive panel indicator LEDs connected to the Main IO and Aux IO lines. The panel indication shown on any specific IO line reflects the state of the Servo in the same group. Thus when using panel indicators such as the Tam Valley Depot Fascia Controller, it is important to use an IO line in the same Group as the Servo being controlled so that the position, movement and lock indications are meaningful.

2 Quick Start

2.1 Connecting Power

The Quad-LN requires an external DC power supply between +5V and +24V. Do not exceed +24V DC. The power input is via the connector with two screw terminals near the upper left corner of the board. Connect the power supply **minus** lead to the **upper** terminal (the one closest to the corner of the board) and connect the power supply **plus** lead to the **lower** terminal (the one furthest from the corner of the board). If the leads are connected backwards, the Quad-LN will not function but will not be damaged.

2.2 Setup without JMRI

All device behavior is controlled via CV values so it is possible to fully configure the device manually. Generally this requires deciphering the CV tables in Section 8 below and manually calculating the proper values, however the following steps will get your Quad-LN up and running controlling turnouts.

2.2.1 Servo Start Address

Set the Servo Start Address first to prevent conflicts with other stationary decoders. (Skip this step if your device is not connected to Loconet).

- Install the **ADDR** jumper. The **ACTIVE** and **MSG** LEDs on the board should start alternately flashing.
- Using your throttle issue a Switch command for any address within the desired block of 8 addresses for the board. When the command is received, the address is set and the LEDs stop flashing.
- Remove the **ADDR** jumper.

2.2.2 Manual Alignment of Travel, Speed and Orientation

The travel endpoints, speed and orientation for each servo can be set manually using the **Select**, **Up** and **Down** buttons on the Remote Alignment Board. Here is the adjustment procedure.

Connect the remote alignment board to the Quad-LN. Remove the Quad-LN **ADDR** jumper if it is installed.

1. Hold the Select button down until a Servo selection LED turns on (about 1 second), indicating that you are in Servo Selection mode and that indicated Servo is selected. (The initial Servo selection will be the same as the one selected the last time alignment was performed.)
2. Use the Up and Down buttons to select the desired Servo, then press the Select button briefly.
3. The Closed LED is now flashing to indicate that the Closed Position is being adjusted. Use the Up and Down buttons to move the Servo to the desired Closed Position. When satisfied with the Closed Position, press the Select button briefly.
4. The Thrown LED is now flashing to indicate that the Thrown Position is being adjusted. Use the Up and Down buttons to move the Servo to the desired Thrown Position. When satisfied with the Thrown Position, press the Select button briefly.
5. The Speed LED is now flashing to indicate that the Speed is being adjusted. Use the Up and Down buttons to change the speed. The Servo will change position to demonstrate the new

Speed. You can continue to adjust the speed while the Servo is moving if desired. When satisfied with the Speed, press the Select button briefly.

6. The Orientation LED is now flashing to indicate that the Closed/Thrown Orientation is being adjusted. If the Closed/Thrown state of the Servo does not match the physical Orientation of the turnout, press either the Up or Down button to reverse the Servo endpoints. This Servo changes positions but the Closed/Thrown state does not change.
7. To readjust any of the parameters, press the Select button briefly to return to step 4.

To exit the Adjustment process at any point hold the Select button until the remote align board LEDs turn off (about 1 second). The new settings are now saved and in use.

Note: the **ACTIVE** or **MSG** LED may light while pressing buttons on the Remote Alignment Board. This is normal.

2.2.3 Automatic Alignment of Travel

The travel endpoints for each servo can be automatically set based upon a user defined AutoAlign Current threshold. The higher the current setting, the higher the force that will be applied at the endpoint. This procedure uses the **Select**, **Up** and **Down** buttons on the Remote Alignment Board. Here is the adjustment procedure.

1. Connect the remote alignment board to the Quad-LN. Be sure to remove the ADDR jumper if it is installed.
2. Move the RUN / ALIGN jumper to the ALIGN position.
3. Hold the Select button down until a Servo Selection LED turns on (about 1 second), indicating that you are in Servo Selection mode and that the indicated Servo is selected. (The initial Servo selection will be the same as the one selected the last time alignment was performed.)
4. Use the Up and Down buttons to select the desired Servo, then press the Select button briefly.
5. If the AutoAlign Current CV is 0 (default, see Section 6.6 for more info), LED3 is now flashing to indicate that the current threshold is being adjusted and it is set to mid-level. Use the Up and Down buttons to selected the desired preset current value. When satisfied with the value, press the Select button briefly.
   If the AutoAlign Current CV is non-zero, that value is used for the current threshold and the procedure moves immediately to the next step.
   Note: To avoid over stressing the device being driven by the Servo, start with lower current threshold values and work up to higher values is needed.
6. The Closed LED is now flashing to indicate that the Closed Position is being adjusted. The Quad-LN moves the Servo to the center position and from there moves it clockwise very slowly while monitoring the Servo current. When the AutoAlign Current threshold is reached, the Closed LED stops flashing and final Closed Position adjustments are made.
7. The Thrown LED is now flashing to indicate that the Thrown Position is being adjusted. The Quad-LN moves the Servo to the center position and from there moves it counterclockwise very

slowly while monitoring the Servo current. When the AutoAlign Current threshold is reached, the Thrown LED stops flashing and final Thrown Position adjustments are made.

8. The new Closed and Thrown positions are stored, the Speed LED lights briefly to indicate the alignment was successful, and then the Quad-LN returns to normal operation.

To abort the Adjustment process at any point, press the Select button until the remote align board LEDs turn off. The prior settings are restored and the Quad-LN returns to normal operation.

Always return the RUN / ALIGN jumper to the RUN position when done.

Note: the **ACTIVE** or **MSG** LED may light when pressing buttons on the Remote Alignment Board. This is normal.

2.2.4 Orientation

If you are connected to Loconet, you can also adjust the Closed/Thrown state of a Servo to match the physical Orientation of the Servo installation by using Loconet Switch commands.

- Install the **ADDR** jumper. The **ACTIVE** and **MSG** LEDs start flashing alternately.
- Using your throttle issue a Switch command. Set the address set to match the desired Servo and select Closed or Thrown based on the physical Orientation of the Servo. When the command is received, the Servo state is adjusted to match the command and the LEDs stop flashing.
- Remove the **ADDR** jumper.

2.2.5 Device Address

The Device Address can be changed using **Service Mode** programming messages that are received by the Quad-LN over Loconet. (No connection between the Quad-LN and the programming track is required.) Choose a unique address between 1 and 16383. See 1.7.1 for information about Device Addresses. A unique 5 digit address is recommended to avoid conflicts.

- Remove all locomotives from the programming track.
- Install the **ADDR** jumper. The **ACTIVE** and **MSG** LEDs should start flashing alternately.
- Enter Programming mode and select Paged mode.
- Write the high byte (upper 8 bits) of the desired address to CV17.
- Write the low byte (lower 8 bits) of the desired address to CV18.
- Remove the **ADDR** jumper.

2.3 Setup with JMRI

The Quad-LN template in JMRI provides the easiest and quickest way to configure the overall device. This section covers the basics needed to get the decoder up and running. Later sections dive into the details.

2.3.1 Create roster entry and set the Device Address

Each decoder needs its own Roster entry so that there is a place to store the configuration settings that are unique to that decoder. In this step we will create the Roster entry for the decoder. We will also assign a unique Device Address to the decoder using the **Service-Mode** Programmer so that we can later alter its configuration whenever required without affecting any other devices. When setting all other parameters of the Quad-LN we will always use the **Ops-Mode** Programmer. The **Service-Mode** Programmer is only used in this step and when performing a Factory Reset.

The Device Address can be set using **Service Mode** Programming messages that are received by the Quad-LN over Loconet. (No connection between the Quad-LN and the programming track is required.)

- Remove all locomotives from the programming track.
- Go to **Programmers** on the **Tools** tab and open the **Service-Mode** Programmer.
- Scroll down the decoder list to **Tam Valley Depot**, expand it, select the **Quad-LN** and then click **Open Programmer**. (see the note below if the entry is missing)
- On the **Roster Entry** tab, enter an ID for the device such as QLNnnnnn (where nnnnn is the 5 digit address for this device) or other unique identifier.
- Go to the **Basic** tab.
- Select **Use Long Address**.
- Enter the desired Device Address in the **Long Address** field. Choose a unique address between 1 and 16383. See 1.7.1 for information about Device Addresses. A unique 5 digit address is recommended to avoid conflicts.
- Click **Save** on the **File** menu to save the new roster entry.
- Go to the **CVs** tab.
- Install the **ADDR** jumper. The **ACTIVE** and **MSG** LEDs should start flashing alternately.
- **Write** CV17 and CV18. (Ignore the "Programmer error" message, this is normal.)
- Remove the **ADDR** jumper.
- **Close** the **Service-Mode** **Programmer**.

Note: If the Tam Valley Depot or Quad-LN entries are missing from the decoder list, please follow these steps to download and install the template file.

- Download the latest template from the /Decoder files/Tam Valley Depot/ folder in the files area of the Yahoo JMRI Users Group. Place the file in the Decoder subfolder under the User File Locations specified in your JMRI Preferences File Locations.
- **Close** the **Service-Mode** **Programmer** if it is still open.
- Select **Recreate Decoder Index** on the **Debug** menu.
- When that step is complete, go back to the first step in this section.

2.3.2 Servo Start Address and Sensor Start Address

Now we are ready to enter all other parameters of the Quad-LN. To do this in JMRI we use the **Ops-Mode** Programmer.

- Go to **Programmers** on the **Tools** tab and open the **Ops-Mode Programmer**.
- Select the Quad-LN entry you created above from the roster and Open it.
- Go to the **Quad-LN** tab.
- Enter the desired servo start address in the **Desired Servo Start Address** field.
- (optional) Enter the desired sensor start address in the **Desired Sensor Start Address** field.
- Click **Write changes on sheet** to update the Quad-LN.
- Click **Save** on the **File** menu to update the roster entry.

2.3.3 Travel and Speed

You can follow the procedure in 2.2.2 above or at any time just use the **Ops-Mode** Programmer in JMRI, click on one of the Group tabs and begin exploring using the information starting in section 3.

2.4 Factory Reset

The Quad-LN settings can always be restored to their factory defaults. Factory Reset is performed by writing a special value to CV 8 (Manufacturer ID) using **Service Mode** programming messages that are received by the Quad-LN over Loconet. (No connection between the Quad-LN and the programming track is required.) Menu choices have been provided in the JMRI **Quad-LN** template file to make this easy to do using the JMRI **Service-Mode Programmer**. Alternate manual methods that do not require JMRI are also described below.

2.4.1 Restore Selective Parameters to Factory Defaults using JMRI

- Remove all locomotives from the programming track.
- Go to **Programmers** on the **Tools** tab and open the **Service-Mode Programmer**.
- Scroll down the decoder list to **Tam Valley Depot**, select the **Quad-LN** and then click **Open Programmer**.
- Select the **Reset** menu and then click **Factory Reset…**
- Select **Restore All CVs…**, **Restore Addresses…** or **Restore Servo and Input CVs...** to specify which parameters to restore.
- Install the **ADDR** jumper. The **ACTIVE** and **MSG** LEDs should start flashing alternately.
- Click **OK** to perform the reset.
- Remove the **ADDR** jumper.
- **Close** the **Service-Mode Programmer**.
- If you chose “Restore Addresses...” above, cycle power to the Quad-LN so the changes can take effect.

2.4.2 Restore All Parameters to Factory Defaults using the Remote Align Board

- Remove power from the **Quad-LN**.
- Install the **ADDR** jumper.
- Connect the remote alignment board to the Quad-LN.
- Press and hold both the Up and Down buttons.
- Reapply power to the **Quad-LN**.

- Continue holding both the Up and Down buttons until the **ACTIVE** and **MSG** LEDs flash alternately for 1 second and then stop flashing.
- Release the buttons.
- Remove the **ADDR** jumper.

2.4.3 Restore Selective Parameters to Factory Defaults without using JMRI

- Remove all locomotives from the programming track.
- Install the **ADDR** jumper. The **ACTIVE** and **MSG** LEDs should start flashing alternately.
- Enter Program Mode on your throttle.
- Write one of the following values to CV8 to specify which parameters to restore.
    - 8 to restore all address, servo and input parameters to their factory default values.
    - 9 to restore all address parameters to their factory default values.
    - 10 to restore all servo and input parameters to their factory default values.
- Remove the **ADDR** jumper.
- Exit Program Mode on your throttle.
- If you wrote 9 to CV8 above to restore only the addresses, cycle power to the Quad-LN so the changes can take effect.

3 Servo Control

To access the following parameters in JMRI always use the **Ops-Mode** Programmer.

3.1 Travel

The Travel parameters allow the Servo Closed and Thrown Position endpoints, Speed, RapidStart feature, and the physical Orientation to be set for each Servo.

3.1.1 Closed Position

The Closed Position endpoint can be set from 0 to 2400, where 0 is fully counter-clockwise and 2400 is fully clockwise.

3.1.2 Thrown Position

The Thrown Position endpoint can be set from 0 to 2400, where 0 is fully counter-clockwise and 2400 is fully clockwise.

3.1.3 Orientation

There might be occasions where the servo Closed and Thrown positions need to be swapped. For example the endpoints might have been accidently reversed during manual configuration, or perhaps an external signal is being used to control a servo and the sense of that signal is opposite what was expected. In these cases the Orientation parameter can be used to swap the travel endpoints so that the Servo Closed/Thrown position indicator and Loconet Switch Closed and Switch Thrown commands work as expected. This is the parameter that is set during the procedure described in 2.2.3. The Orientation parameter was implemented mainly to save users that do not have JMRI available from

having to repeat the process of manually setting the Closed Position and Thrown Position endpoints just to reverse the endpoints.

The available Orientation settings are **Normal Endpoints** and **Reverse Endpoints**. JMRI users are free to either use the Orientation parameter or easily obtain the equivalent effect by swapping the endpoint values themselves.

3.1.4 Speed

The Speed can be set from 0 to 63, where 0 is very slow and 63 is very fast. A typical Speed setting is 4. When you reduce the speed of other types of slow motion switch machines by lowering the drive voltage, this also reduces the applied force (torque). With servos the position is controlled continually during movement so full torque is available. This makes movement smooth even at very slow speed settings.

3.1.5 RapidStart

With any slow motion turnout motor there is generally some amount of over-travel to ensure that the throwbar is tensioned and the point rail is locked against the stock rail. When the motor is reversed, no movement of the points occurs until the over-travel is "unwound" and the motor begins to drive the throwbar in the opposite direction. The time it takes to unwind the over-travel is noticeable at slow motor speeds and detracts from the realistic movement that slow motion motors are intended to create. In some cases this effect creates the impression that the turnout action requested by the operator failed to occur, and results in operators re-actuating turnout control buttons and toggles.

The RapidStart feature is designed to minimize the over-travel delay and provide more realistic throwbar motion. When RapidStart is enabled and a servo is reversed, the Quad-LN moves the servo through the first part of the range of motion at high speed and then switches to the speed set by the Speed parameter for the remainder of the motion. The initial movement at high speed quickly unwinds the over-travel so that the throwbar beings moving much sooner, while the final movement at low speed creates the desired slow motion effect.

The RapidStart parameter allows you to set the fraction of the movement that occurs at high speed. The available RapidStart settings are **none**, **one-eighth travel**, **one-quarter travel**, and **three-eighths travel**.

3.1.6 Directional Speed

Normally the Quad-LN drives a servo at the same speed whether the servo is moving towards the Closed position or towards the Thrown position. The Quad-LN also allows each servo to have independent Speed (and RapidStart) settings for each direction of movement. This feature might be useful when using a servo to drive a semaphore blade or an animation function like a pile driver, where the speed in one direction is different than the other.

The Directional Speed parameter allows you to set independent Speed and RapidStart values for each direction of travel. The available Directional Speed settings are **Disable** and **Enable**. When Directional

Speed is set to **Enable**, the JMRI template will display two additional settings: “Thrown Speed” and “Thrown RapidStart”.

3.1.7 Thrown Speed

When Directional Speed is enabled, the Thrown Speed parameter sets the speed when moving in the Thrown direction. The Speed parameter in 3.1.4 above always sets the speed when moving in the Closed direction.

3.1.8 Thrown RapidStart

When Directional Speed is enabled, the Thrown RapidStart parameter sets the RapidStart behavior in the Thrown direction. The RapidStart parameter in 3.1.5 above always sets the behavior in the Closed direction.

3.2 Lock

In systems like CTC, some turnouts are controlled remotely by the dispatcher. These remotely control turnouts are normally “locked “, that is they cannot be operated by a local train crew member. When a train crew wishes to operate one of these controlled turnouts such as during local switch moves, they are required to first obtain permission from the dispatcher before the turnout is “unlocked” and then operated locally.

Besides supporting realistic operation under CTC, the ability to lock a turnout is useful just to prevent it from changing inadvertently. For example, if running a train continuously during an Open House it is desirable to lock the mainline route so that someone bumping into a fascia control doesn’t cause a derailment or worse.

The Quad-LN Lock feature permits local control of a turnout to be locked out and has the ability to provide a local indication of the state of the turnout lock. Locking a Quad-LN Servo means that local control actions via the Main and Aux IO lines (i.e. buttons, toggles and secondary triggers) can no longer directly change the Servo position. The Servo still responds normally to Loconet Switch commands at its address, including commands originated by a secondary messages or by routes.

The Loconet Switches at addresses **Servo Start Address** + 4 to **Servo Start Address** + 7 control the locks on Servo 1 to Servo 4 respectively. When a Lock Control Switch is Thrown, the corresponding Servo is locked, and when the Lock Control Switch is Closed, the corresponding Servo is unlocked.

The available Lock settings for each Servo are **Not Lockable and Local Lockout**. When set to **Not Lockable**, the local controls are not affected by the state of the Lock Control Switch. When set to **Local Lockout**, the local controls for the turnout can be locked out by setting the Lock Control Switch to Thrown.

Example 1. Servo 1 Lock Mode is set to **Local Lockout** and the Quad-LN Servo Start Address is set to 9. This means Servo 1 is at Loconet Switch address 9 and the Lock Control Switch for Servo 1 is at Loconet

Switch address 13. When Switch 13 is Thrown, local control of Servo 1 is locked, and when Switch 13 is Closed local control of Servo 1 is unlocked.

Example 2. Servo 1 Lock Mode is set to **Not Lockable** and the Quad-LN Servo Start Address is set to 9. Changing Switch 13 to Thrown or Closed has no effect on local control of Servo.

The Quad-LN Lock mechanism supports unlocking either by the local crew or by the Dispatcher as desired. The local crew could control the lock by sending the appropriate Switch command to the Lock Control Switch using a handheld throttle. A more interesting approach might be for the local crew to control the lock via a keyswitch or other special electrical interlock device that is connected to a Quad-LN input pin. There is an example in 4.4.5 showing how to accomplish this. Having the local crew control the lock in this manner is similar to prototype practice but requires hardware like a keyswitch for every turnout to be unlocked.

An alternative approach is to have the Dispatcher control the lock by sending the appropriate Loconet Switch command, which could be done via a control panel lever for instance. This approach doesn't require extra hardware and the Tam Valley Fascia Controller can indicate the state of the Lock to the local crew to avoid confusion (see 4.2.1).

To lock specific turnouts to protect a continuously running train during an open house, the Lock Control Switch for any turnout can be set using a handheld throttle. Lock Control Switches can be part of a Route, making it possible to lock or unlock a series of turnouts with just a single action. The Lock or Unlock Routes could even be fired by a pushbutton or toggle switch connected to one of the Quad-LN inputs, making the process of locking or unlocking key fascia turnout controls very simple. There is an example in 4.4.5 showing how to configure a pushbutton press to fire a Route. Tam Valley Fascia Controllers can display the state of the Lock so it is easy to see that the layout is secured (see 4.2.1).

3.3 Midpoint

Midpoint is a special mode that causes the servo to move midway between the Closed and Thrown setpoints when the Lock Control Switch is Thrown. This mode may be useful for control of 3 position semaphores, 3-way stub turnouts, and special animation effects.

The available Midpoint settings for each Servo are **Disable** and **Enable**. When set to **Disable**, the Servo position is not affected by the state of the Lock Control Switch. When set to **Enable**, the Servo will move to the mid-travel position when the Lock Control Switch is set to Thrown and will return to the last commanded position when the Lock Control is set to Closed.

Both LEDs on a Tam Valley Depot Fascia Controller will be lit when the corresponding servo is at mid position. If using bicolor LEDs, the Red and Green elements are switched on and off rapidly so that each is on half the time, producing a shade of Yellow. The switch rate is 100 times/second which is higher than the eye can detect so no flicker is visible. If using a custom circuit to drive an indicator, the

Yellow shade can be varied by changing the Red and Green LED current levels (often more Red current / less Green current is required).

3.4 Message - Servo Position Feedback without sensors

The Quad-LN sends continuous position commands to each servo, so it always knows the servo position. As a result the Quad-LN can provide Turnout Position Feedback messages based on the servo position. Unless you really require physical turnout point position feedback, this feature is a very effective way to provide visual indication of the turnout state on your dispatcher panel without any sensors!

The available Message choices are **No Output Feedback**, **Output level feedback**, and **Exact feedback (sim)**. If using a JMRI control panel, generally **Exact feedback (sim)** is a better choice than **Output level feedback** but there are currently tradeoffs. See Section 9 for a brief technical discussion of how JMRI v 2.14 handles switch feedback and the pros and cons of each feedback choice. (A future JMRI release is expected to eliminate any drawbacks of Exact Feedback.) If you are not using software that monitors turnout position feedback or you just don't require them for your layout, then select **No Output Feedback** to eliminate the unneeded Loconet messages.

3.4.1 Exact feedback (sim)

Exact feedback normally utilizes two limit switches that are mounted on the turnout: one limit switch turns on when the turnout reaches the Closed position and the other limit switch turns on when the turnout reaches the Thrown position. Feedback messages are sent when either limit switch changes state. In Exact feedback (sim) mode, the Quad-LN sends the same feedback messages but bases them on the servo position. When the Servo starts to move the limit switch corresponding to the current position is reported as Low and when the Servo reaches its endpoint the limit switch corresponding to the final position is report as High.

3.4.2 Output level feedback

With traditional switch motors Output level feedback shows whether the Closed output driver is On (High) or the Thrown output driver is On (High). This information conveys the direction the switch motor is being driven but not whether the travel has been completed. The Quad-LN modifies this report to mimic the approach used in Exact feedback. The appropriate output driver is still reported as High once the travel endpoint is reached, but during movement both output drivers are reported as Low.

3.5 Cascade

Cascade action refers to a condition where command to one turnout also triggers a command to a second turnout. If the second turnout triggers a command to a third turnout, etc, the command is said to "Cascade" down. Cascade action provides a very handy way to align a route through a yard ladder or crossover by sending only one Loconet Switch command.

The details of the Cascade settings are shown below, followed by a couple examples to illustrate the basics.

3.5.1 Trigger

The Trigger is the specific Switch command that initiates the Cascade event. When the Trigger command is received, the Cascade Switch command is sent.

The available Cascade Trigger settings for each Servo are **None, Closed, Thrown** and **Closed or Thrown**. When set to **Closed**, the Cascade action is triggered whenever the Servo receives a Closed Switch command. When set to **Thrown**, the Cascade action is triggered whenever the Servo receives a Thrown Switch command. When set to **Closed or Thrown**, the Cascade action is triggered whenever the Servo receives any Switch command.

3.5.2 Action

The Action is the type of Switch command that is sent to the Cascade turnout when the Trigger condition is met.

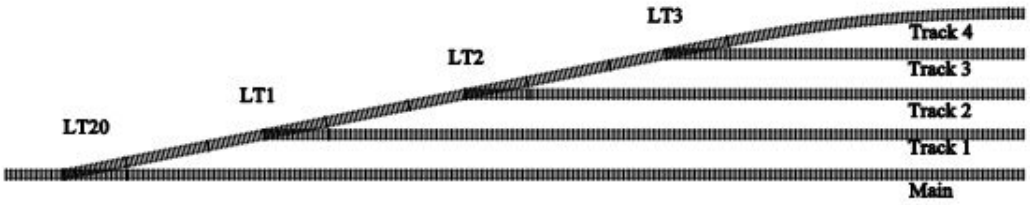
The available Cascade Action settings for each Servo are **Closed, Thrown, Follow** and **Invert**. When set to **Closed**, a Closed Switch command is sent. When set to **Thrown**, a Thrown Switch command is sent. When set to **Follow**, a Switch command matching the state of the Servo is sent. When set to **Invert**, a Switch command opposite to the state of the Servo is sent

3.5.3 Turnout

The Turnout is the Switch number for the turnout that is to receive the Cascade Switch command.

3.5.4 Cascade Examples

Yard ladder alignment. We have a number of yard tracks accessed via a simple ladder.



The entry to the yard from the main is controlled by Switch 20: Closed for the main and Thrown for the ladder. Switches on the yard ladder are Closed for the ladder and Thrown to select their specific track, so Switch 1 is Thrown for track 1, Switch 2 is Thrown for track 2, etc. Suppose we wish to fully line the route from the main to any yard track by only issuing one Switch command.

First set Switch 1 Cascade Trigger to **Closed or Thrown**, Cascade Action to **Thrown**, and Cascade Turnout to **20**. Now whenever Switch 1 receives any command, it will send a message to Switch 20 to move to the Thrown position. To line a train into Track 1, all we need to do is set Switch 1 to Thrown: Switch 20 will get Thrown by the Cascade action.

Next set Switch 2 Cascade Trigger to **Closed or Thrown**, Cascade Action to **Closed**, and Cascade Turnout to **1**. Now whenever Switch 2 receives any command, it will send a message to Switch 1 to move to the Closed position. Switch 1 will in turn send a command to Switch 20 to move to the Thrown position. To line a train into Track 2, all we need to do is set Switch 2 to Thrown: Switch 1 will get Closed and Switch 20 will get Thrown by the Cascade action.

Repeat this process for each of the remaining tracks. The key is that any change to the turnout that directly selects the track sends a Cascade message to the preceding turnout that aligns it as required.

Crossover. We have a crossover is controlled by two turnouts at addresses 6 and 15.

For the normal routes, both 6 and 15 are Closed. For the crossover route, both 6 and 15 are Thrown. Now suppose we want to just send Switch 6 Closed and Switch 6 Thrown to select the routes.

One simple way for this to happen is if Switch 6 sends a Cascade message to Switch 15 to Follow it anytime Switch 6 is commanded to a position. Set Switch 6 Cascade Trigger to **Closed or Thrown**, Cascade Action to **Follow**, and Cascade Turnout to **15**. Now whenever Switch 6 receives a command, it will send a message to Switch 15 to move to the same position.

4 Main and Aux IO lines

Each Quad-LN Servo has an associated Main IO line and Aux IO line. The Quad-LN samples each input line hundreds of times per second and debounces (filters) input changes to suppress switch bounce and electrical noise. The simplest use of the input capability of an IO line is to control its associate Servo. More complex actions are also possible, for instance a single IO line can control all four of the Quad-LN servos. The specific behavior of each input is configured using the parameters described in this section. To access the following parameters in JMRI always use the **Ops-Mode** Programmer.

4.1 Input Response

4.1.1 Trigger

Input actions are initiated only when an input change occurs that matches the Trigger condition.

The available Input Trigger settings for each Input are **None, Positive Edge, Negative Edge** and **Both Edges**. When set to **Positive Edge**, the input action is triggered whenever the Input changes from the Low level to the High level. When set to **Negative Edge**, the input action is triggered whenever the

Input changes from the High level to the Low level. When set to **Both Edges**, the input action is triggered whenever any Input change occurs. When set to **None**, no action is triggered by the input and the input line is not sampled.

4.1.2 DCC Freeze

Some input devices give false indications when DCC is not present. For example, a short in a DCC power district may cause all block occupancy detectors in that district to indicate that their blocks are vacant whether or not trains are present. The Quad-LN has the ability to monitor the DCC signal and to freeze an input at its previous level when the DCC signal is lost, thus prevent false reports from detectors during short circuits.

The available settings are **Always Live** and **Freeze on DCC loss**. When set to **Always Live**, input action follows the Trigger setting independent of the DCC state. This is generally the appropriate setting when the input is used for Servo control. When set to **Freeze on DCC lost**, input action is disabled when no DCC signal is present on the DCC input pins. This is generally the appropriate setting when the input is used for a block occupancy detector.

4.1.3 Message

The Quad-LN can send a Loconet message when the input Trigger condition is met. This permits the inputs to be used for detectors, turnout feedback, and other purposes.

For Main IO lines, the available settings are **General Sensor** and **Thrown Position Feedback.** For Aux IO lines, the available settings are **General Sensor** and **Closed Position Feedback.**

When set to **General Sensor**, a Sensor message is sent. When set to **Closed** or **Thrown Position Feedback**, the corresponding Turnout Sensor Report is sent.

4.2 Indication

When the Input line is not being sampled, the Quad-LN has the capability to treat the line as an Output and can drive an LED indicator to display the state of the Servo. Since the input sampling takes place very quickly and at a rate higher than the eye can detect, the LED indicator will appear to be on continuously with no discernible flicker. (If using input devices other than Tam Valley Depot Fascia Controllers, please see the general Quad datasheet for information on simultaneously wiring switches and LEDs to the Input lines.)

4.2.1 LED Mode

When the LED indicator is enabled, the steady LED indication shows whether the Servo is Closed or Thrown. The LED can be set to blink rapidly while the Servo is moving, thus providing feedback that travel is complete when the indicator again becomes steady. When the associated Servo is Lockable, the LED can be set to blink slowly to provide feedback on the whether local control of the Servo is Locked or Unlocked. (The global Lock Indicator setting determines whether blinking indicates the Locked or the Unlocked state - see 6.8.)

The available LED Mode settings for each Input are **Disable – Input only, Steady Drive, Blink on Move, Blink on Move or Lock, Blink on Lock** and **Lock State**. When set to **Steady Drive**, the output indication shows the commanded Servo position. When set to **Blink on Move**, the output indication shows the commanded Servo position and blinks rapidly while the Servo is moving. When set to **Blink on Move or Lock**, the output indication shows the commanded Servo position, blinks rapidly while the Servo is moving and blinks slowly when the Servo is **Lockable** and the state of the lock matches the Lock Indicator setting. When set to **Blink on Lock**, the output indication shows the commanded Servo position and blinks slowly when the Servo is **Lockable** and the state of the lock matches the Lock Indicator setting. When set to **Lock State**, the output indication shows the commanded Lock position instead of the Servo position. When set to **Disable – Input only**, no output drive is provided and the line is always left as an input. This is appropriate when the input device is a detector for example.

4.2.2 LED Sense

If the LED panel wiring results in the opposite indication from what is desired, the Quad-LN can simply invert the output drive to the LED. This eliminates the need to rewire the LED indicators or remount the Fascia Controller.

The available LED Sense settings for each Input are **Normal** and **Inverted**. When set to **Normal**, the LED output drive is High when the Servo is Closed and Low when Thrown. When set to **Inverted**, the LED output drive is Low when the Servo is Closed and High when Thrown.

4.3 Action

Each Quad-LN input can control any or all of the Servos. This allows inputs to trigger local routes without requiring a Loconet connection. The local action of any input on each of the four Quad-LN Servos can be set independently. (The local action on a Servo is only performed if that Servo is not locked.)

The available Servo action settings for each Input are **None, Toggle, Follow, Invert, Closed, Thrown, Lock Toggle, Lock Follow, Lock Invert, Lock Closed and Lock Thrown**. When set to **None**, the state of the corresponding Servo is unaffected by the input. When set to **Toggle**, the state of the corresponding Servo is set to the opposite of its current state. When set to **Follow**, the state of the corresponding Servo is set to Closed if the Input is High and set to Thrown if the Input is Low. When set to **Invert**, the state of the corresponding Servo is set to Closed if the Input is Low and set to Thrown if the Input is High. When set to **Closed**, the state of the corresponding Servo is set to Closed. When set to **Thrown**, the state of the corresponding Servo is set to Thrown. When one of the **Lock** settings is selected, the state of the corresponding Lock is set accordingly.

4.4 Secondary Message

Each Input can receive or send a Secondary message. This can be used to implement more complex logic for yards, panels or other functions. If set to send a Secondary message, when the Input Trigger condition set in 4.1.1 is met the Secondary message is sent. If set to receive a Secondary message, a Loconet sensor or turnout message that matches the Secondary Trigger fires the Servo Actions that are configured for the Input.

4.4.1 Type

The available Secondary Type settings for each Input are **Trigger on this message** and **Send this message**. When set to **Trigger on this message**, the specified Servo Actions for the input are executed when the Condition below is met. When set to **Send this message**, the Secondary message is sent when the Input Trigger condition is met, and the message content is determined by the Condition below.

4.4.2 Device

The available Secondary Device settings for each Input are **Turnout** and **Sensor**. When set to **Turnout**, the Secondary message is a Switch command message. When set to **Sensor**, the Secondary message is a General Sensor message.

4.4.3 Condition

The available Secondary Condition settings for each Input are **None, Hi / Closed, Lo / Thrown** and **Both / Follow**.

If Type is **Trigger on this message:** When set to **Hi / Closed**, the trigger condition is a Sensor Hi or Switch Closed message. When set to **Lo / Closed**, the trigger condition is a Sensor Lo or Switch Thrown message. When set to **Both / Follow**, the trigger condition is a Sensor or Switch message in either state. When set to **None**, the Secondary message function is disabled.

If Type is **Send this message:** When set to **Hi / Closed**, the message sent is a Sensor Hi or Switch Closed message. When set to **Lo / Closed**, the message sent is a Sensor Lo or Switch Thrown message. When set to **Both / Follow**, the message sent is a Sensor or Switch message with the state matching the Input state. When set to **None**, the Secondary message function is disabled.

4.4.4 Number

The Switch or Sensor number for the Secondary message.

4.4.5 Secondary Message Examples

All Mainline Turnouts to Closed. Suppose we have many mainline turnouts controlled by Quad-LN boards. Each turnout is controlled by a pushbutton on its Main IO line. We would like to be able to set all the turnouts to their Closed position with a single Switch command, say Switch 200 Closed. For each Servo Main IO line, set the Secondary Message Type to **Trigger on this message**, the Device to **Turnout**, the Condition to **Hi / Closed** , and the Number to **200**. Now sending a single Switch command lines all the turnouts to the Closed position.

All Mainline Turnouts to Closed with a single button press. Same as the previous example, except we want to press a pushbutton to line the turnouts instead of sending a Switch command with a throttle. Do the setup as in the previous example. Wire the pushbutton to an available Aux IO line on a Quad-LN board, then adjust the settings for the Aux IO line as follows: Trigger to **Positive Edge**; Action to **None** for each Servo; Secondary Message Type to **Send this message**, Device to **Turnout**, Condition to **Hi/Closed** and Number to **200**. Now pressing the button sends a Switch 200 Closed command, which in turn causes all the mainline turnouts to move to the Closed position.

Unlock a Mainline Turnout using a keyswitch. We have a local fascia pushbutton set up for control of Switch 17, but wish to lock (disable) the pushbutton unless a keyswitch is actuated. Wire the keyswitch to an available IO line on a Quad-LN board, then adjust the settings for the IO line as follows: Trigger to **Both Edges**; Action to **None** for each Servo; Secondary Message Type to **Send this message**, Device to **Turnout**, Condition to **Both/Follow** and Number to 21 (21 is the Lock Control Switch for Switch 17 – see 3.2). Now opening the keyswitch sends a Switch 21 Closed command which unlocks Switch 17, and closing the keyswitch sends a Switch 21 Thrown command which locks Switch 17. A suitable indicator on the Dispatcher panel could monitor Switch 21 and provide an indication when Switch 17 is unlocked.

5 Routes

Routes are arbitrary sequences of Switch and Sensor messages that are sent when a trigger condition is met. The Quad-LN has 4 routes, each containing up to 8 entries consisting of Switch command or Sensor state messages. Routes can be set to Expand the prior route, so it is possible to have a single route with as many as 32 entries. This makes it easy to do things like create an arbitrary track alignment by sending a single command. To access the following parameters in JMRI always use the **Ops-Mode** Programmer.

5.1 Type

The available Route Types are **None, Normal Route, Selector** and **Expand (Prior) Route**. A **Normal Route** is only triggered by the first entry and always sends the remaining entries exactly as entered. A **Selector Route** is triggered by any of the entries and sends the remaining entries in the opposite state from that entered. This action implements a "1-of-n" selection. An **Expanded Route** simply adds its entries to the prior route.

5.2 Entries

Each Route entry consists of a Device Address and Device Type / Action. For Switch messages the available Actions are **Closed** and **Thrown**, for Sensor messages the available Actions are **Sensor Hi** and **Sensor Lo**.

5.3 Route Examples

Align a Route when a button is pressed. We would like to be able to line a route that involves many turnouts by pressing a pushbutton. Wire the pushbutton to an available Aux IO line on a Quad-LN board, then adjust the settings for the Aux IO line so that the Message is set to **General Sensor**. Set the Action to **None** for each Servo. Pick a Route in any Quad-LN and set the Type to **Normal Route**, set the first entry to the **Aux IO address** and **Sensor Hi**, and then enter the **Switch address** and **Closed** or **Thrown** in the following entries for each turnout in the route. Use Expanded Routes as needed. Now pressing the button lines the entire route.

Lock All Mainline Turnouts. We have many mainline turnouts controlled by Quad-LN boards. Each turnout is controlled by a pushbutton on its Main IO line. We would like to be able to lock out all the pushbuttons (for continuous running during an Open House for example) by sending a single Switch command, say Switch 200 Closed. Pick a Route in any Quad-LN and set the Type to **Normal Route**, set the first entry to **200 Closed**, and then enter the **Lock address** and **Thrown** in the following entries for each mainline turnout lock. Use Expanded Routes as needed. Now sending a single Switch command locks all the mainline turnouts.

Panel track selection indicator. We have a yard panel with an LED for each track. The LEDs are controlled via a stationary decoder(s) using Switch commands. Setting a particular Switch address to Closed turns an LED on, and setting it to Thrown turns the LED off. We would like to be able to send a single Switch command to light the LED for the selected track and have all the other LEDs go out. Pick a Route in any Quad-LN, set the Type to **Selector** and then enter the **Switch address** and **Closed** in the route entries for each Switch address that controls an LED. Use Expanded Routes as needed. Now sending a single Switch command turns on the LED for the selected track and turns off the LEDs for the non-selected tracks.

6 The Quad-LN Pane

The Quad-LN has a number of address settings and special configuration settings. Most users will only need to set the Servo Start Address and Sensor Start Address here (the Long Address is shown for information only and will have already been set if required). The remaining special configuration settings can generally be left at their default values. To access the following parameters in JMRI always use the **Ops-Mode** Programmer.

6.1 Addresses

6.1.1 Long Address

The Long Address is the address used when programming the device in **Ops Mode**. This address is also shown on the Basic pane. See Section 1.7.1 for more info. A special procedure is required to change the device address, and this procedure is described in 2.2.5. This procedure is intended to prevent inadvertent device address changes.

6.1.2 Desired Servo Start Address

The Switch addresses for the Quad-LN Servos are set using this field. Since the Quad-LN using 8 consecutive Switch addresses, the value entered here is adjusted down as needed to the group-of-8 boundary. The actual address for each Servo is displayed on its Group tab. See Section 1.7.2 for more info on the Servo Start Address.

6.1.3 Desired Sensor Start Address

The Sensor addresses for the Quad-LN Servos are set using this field. Since the Quad-LN using 8 consecutive Sensor addresses, the value entered here is adjusted down as needed to the group-of-8

boundary. The actual address for each Sensor is displayed on its Group tab. See Section 1.7.3 for more info on the Sensor Start Address.

6.2 Interrogate Options

A Command Station or a computer connected to Loconet can monitor all Loconet messages in order to keep track of state changes, but until a particular Switch or Sensor changes it won't generate a message. So a mechanism to determine the current state of all Switches and Sensors is needed. This mechanism is provided via special Loconet Switch commands that are reserved just to interrogate stationary decoders for their device status. A DCS100 typically sends these special Switch commands every time it receives a Track Power On (GPON) message. Once the responses from all the stationary decoders are collected, the state of all connected devices is known.

The Interrogate options control the way the Quad-LN responds when interrogated. These options can normally be left at their default values.

6.2.1 Input Interrogate

This setting controls whether the Quad-LN reports the state of its Sensors when interrogated. The default value is **Enable**.

6.2.2 GPON Interrogate

This setting controls whether the Quad-LN treats a Track Power On (GPON) command as an interrogate command. The default value is **Disable**, since the DCS100 interrogates all devices after GPON using the special Switch Comands. Setting GPON Interrogate to **Enable** may be useful in cases such as when a computer is monitoring the layout via Loconet but no DCS100 is present.

6.3 Servo State Memory

This setting controls whether the Quad-LN records the Servo positions when they change and then restores all Servos to their last known position at power-up. The default value is **Enable**. If this setting is changed to **Disable** then each Servo is set to Closed at power-up.

6.4 Retry on DCC Busy

This setting controls whether the Quad-LN attempts to retry Switch Requests when the Command Station reports that the DCC bus is busy. The default value is **Disable**.

Note: This section is somewhat technical. Users that don't need any Switch Request sent by the Quad-LN to reach DCC-only stationary decoders can simply ignore this section. The information below is included so that those users with complex configurations that include DCC-only stationary decoders can better understand the system performance tradeoffs.

When the Quad-LN sends a Switch Request on Loconet as part of a Cascade, Secondary Message or Route, the Command Station attempts to send the command over the DCC bus. This is done so that DCC-only stationary decoders will also receive the command. However due to the difference in bandwidths, Loconet Switch Requests can arrive at the Command Station faster than the Command

Station can send them out over DCC. When this happens the Command Station internally buffers the Switch Requests in order to give the DCC bus a chance to catch up. When the Command Station receives a Switch Request and its internal buffer is already full (or with some Command Stations if track power is turned off so that DCC commands cannot be sent), the Command Station responds with a special Loconet message to indicate that the Switch Request did not make it to the DCC bus.

When a Switch Request does not get to the DCC bus, the Quad-LN has two options: ignore the failure or send the Switch Request again. Sending the Switch Request again usually works out fine, but if there are a large number of Switch Requests being are generated on Loconet (from one or more devices) then the retries can start to pile up. When this happens performance degrades until the Command Station is able to get the DCC side caught up, and under most circumstances things get resolved without any noticeable impact.

If you need the Quad-LN to be able to send Switch Requests to DCC-only stationary decoders then set this option to **Enable** and the Quad-LN will retry Switch Requests when required. If you have not set up the Quad-LN to send Switch Requests to DCC-only stationary decoders, then leave this option set to **Disable** since in this case it doesn't matter whether a command sent by the Quad-LN makes it to the DCC bus and thus there is no reason to create additional Loconet traffic by resending Switch Requests.

Finally, if you are using a DCS100 and have no DCC-only stationary decoders at all, you can set OPSW 27 to Closed. Closing OPSW 27 tells the DCS100 not to send any Switch Requests over the DCC bus. This eliminates the entire DCC backlog issue and allows Switch Requests to be sent at full Loconet speed under all conditions.

6.5 Send Output Off Switch Request

This setting controls whether the Quad-LN sends the Output Off form of the Switch Request after sending the Output On form. The default value is **Enable**.

Note: As with the preceding section, this section is somewhat technical and most users can just ignore it.

When the Quad-LN sends a Switch Request on Loconet as part of a Cascade, Secondary Message or Route, two Loconet messages are generated. The first message is an Output On Switch Request and the second is an Output Off Switch Request. By convention these two messages are sent because some stationary decoders turn their output drive on when they receive an Output On Switch Request and turn the drive off when they receive an Output Off Switch Request. The Quad-LN inserts a delay of about 100 milliseconds between the two messages.

When receiving Switch Requests, the Quad-LN ignores the Output Off Switch Requests and leaves the output drive on all the time. Many stall motor decoders act this way. Although sending the Output Off messages doesn't affect the decoders it does double the amount of Loconet traffic and doubles the amount of ***DCC*** traffic (see 6.4). This can reduce performance briefly during intense bursts of Switch

Request activity, such as when sending a long Route. If all your stationary decoders ignore the Output Off Switch Request then setting this option to **Disable** will eliminate the unnecessary Loconet messages.

6.6 Local Action Visibility

This setting controls whether the Quad-LN makes local Servo and Lock actions visible on Loconet. The default value is **Visible on Loconet**.

The Quad-LN can perform the local Servo or Lock actions that are triggered by input changes (see 4.3 for information on local actions) in two slightly different ways. In the first approach the Quad-LN directly changes the Servos and Locks to the new values. In the second approach the Quad-LN generates Loconet Switch Request commands just as would happen if some other other Loconet device were sending the command. The second approach is used by default and provides Loconet visibility to the state changes caused by the local actions the same as with any Loconet Switch Request. If Loconet visibility is not needed, then you can set this option to **Perform Internally** and eliminate the unnecessary Loconet message traffic. Regardless of which setting you choose, the local actions will be performed whether or not the Loconet is connected.

6.7 AutoAlign Current

This setting controls the current threshold used during the Automatic Alignment Procedure (Section 2.2.3). The default value is 0, which means that the user must select the desire current threshold from one of eight preset values each time Automatic Alignment is performed. If a non-zero value is entered for this parameter, then that value will be used each time the Automatic Alignment is performed.

Higher current threshold values equate to greater Servo force at the travel endpoints. Care should be exercised when entering values to ensure the limit is appropriate for the device being driven by the Servo. Start with a lower value and work up to higher values as needed to achieve the desired travel during alignment.

6.8 Lock Indicator

This setting determines whether the IO line LED indicators configured to display their Servo Lock status will blink when the Servo is in the Locked state or when it is in the Unlocked state. The default value is **Blink when Locked**.

6.9 Device Information

This section contains information mainly useful for debugging problems and performing updates.

6.9.1 Manufacturer ID

The Manufacturer ID for Tam Valley Depot is 59. Writing special values to this parameter when in **Service Mode** can reset the device – see Section 2.4.

6.9.2 Product ID

This value identifies the product and helps prevent installing incorrect firmware updates. The Product ID for the Quad-LN is 5.

6.9.3 Hardware Version

This value identifies the product hardware revision and helps prevent installing incorrect firmware updates.

6.9.4 Software Version

This value identifies the product software revision and helps prevent installing incorrect firmware updates.

6.9.5 Build Number

This value is an additional software identifier and enables better factory support.

6.10 Other info for JMRI use only

These values are used for internal calculations and should never be changed by the user.

7 Updating the Firmware

The Quad-LN design incorporates a bootloader that permits the firmware to be updated in the field using JMRI. The bootloader portion of the firmware is protected and cannot be accidentally overwritten during the update process. Normally the Quad-LN user settings are not affected by the firmware update process. However it is always best to ensure your Quad-LN decoder file is up to date before performing a firmware update so that your settings can be easily restored later if required.

To perform a firmware update, follow the steps below.

1. Download the new firmware from TamValleyDepot.com and place the file in your JMRI Preferences folder.
2. Select **Download Firmware** from the JMRI **LocoNet** menu.
3. Click the **16 bit** File format button if it isn't already selected.
4. Click the **Select** button and then open the downloaded file.
5. Click **Read file** to load the file into the downloader.
6. Disconnect power from the Quad-LN.
7. Connect the Remote Alignment Board to the Quad-LN.
8. Hold down the **Select** button on the Remote Alignment Board and reapply power.
9. After the Quad-LN comes up and the **ACTIVE LED** is lit, release the **Select** button. The Quad-LN is now in bootloader mode.
10. In JMRI, click **Download** to begin downloading the file to the Quad-LN. The **MSG LED** will toggle as each message is received. The **Servo 2 LED** will flash as each block of memory is written.
11. If the download completes successfully, the Quad-LN will resume normal operation.
12. If an error occurs, the **ACTIVE LED** will flash about once/second. One of the four **Servo LEDs** will be lit to indicate the error code.

- Error 1 = communication error - bad message received.
- Error 2 = firmware file error - wrong manufacturer code or wrong product code. The file you are attempting to download is not for this product.
- Error 3 = hardware version error – the firmware is not for this version of the board. Obtain the proper firmware file for the hardware version of your board.
- Error 4 = software version error – the firmware isn't newer than the existing firmware. The current firmware is as new or newer than the firmware you are attempting to download. If the download is still desired, select **Don't check software version** in the JMRI Firmware Downloader and repeat the process starting with step 6.

Note: The Quad-LN software is overwritten as the download progresses. Once the download starts, if it is interrupted or if Error 1 occurs then the Quad-LN software will probably be corrupted. However the bootloader will be intact, so just click **Abort** and repeat the download process starting with step 6. In cases where there is a great deal of other Loconet traffic, try increasing the **Delay (msec)** value to 100. This will slow the download process but will make it more tolerant of Loconet collisions.

If Error 2, 3 or 4 occurs, the Quad-LN did not accept the download and the previous software is intact. If you still want to perform the update, resolve the issue and then repeat the download process.

8 CV Tables

8.1 Full CV Table

CV #	Description	Range	Default
1	Short Address		1
3,2	Servo Address Block (Servo Start Address–1)/8	0-254	0 (0,0)
4	Software Version	0-255	-
5	Software Build Number	0-255	-
6	Hardware Revision	0-255	-
7	Product ID	5	5
8	Manufacturer ID	59	59
11,10	Sensor Address Block (Sensor Start Address–1)/8	0-511	0 (0,0)
17,18	Long Address	Table 8-2 Long Address CV Format.	11000 (234, 248)
19	Auto Align Current	0-255	0
21	Configuration	Table 8-3 Configuration CV Format.	47
34,33	Servo 1 Closed Position	0-2400	1260 (4,236)
36,35	Servo 1 Thrown Position	0-2400	1140 (4,116)
37	Servo 1 Speed	Table 8-4 Servo Speed CV Format.	4
38	Servo 1 Output Option	Table 8-5 Servo Output Option CV Format.	2
40,39	Servo 1 Cascade Address	Table 8-6 Servo Cascade Address CV Format.	0 (0,0)
41	Main IO 1 Option	Table 8-7 Main IO Option CV Format	3
43,42	Main IO 1 Action	Table 8-8 Main IO Action CV Format	2
45,44	Main IO 1 Secondary	Table 8-9 Main IO Secondary CV Format.	0 (0,0)

CV #	Description	Range	Default
46	Aux IO 1 Option	Table 8-10 Aux IO Option CV Format	33
48,47	Aux IO 1 Action	Table 8-11 Aux IO Action CV Format	1
50,49	Aux IO 1 Secondary	Table 8-12 Aux IO Secondary CV Format.	0 (0,0)
51	Servo 1 Thrown Speed	Table 8-4 Servo Speed CV Format.	4
52-54	Reserved		
56,55	Servo 2 Closed Position	0-2400	1260 (4,236)
58,57	Servo 2 Thrown Position	0-2400	1140 (4,116)
59	Servo 2 Speed	Table 8-4 Servo Speed CV Format.	4
60	Servo 2 Output Option	Table 8-5 Servo Output Option CV Format.	2
62,61	Servo 2 Cascade Address	Table 8-6 Servo Cascade Address CV Format.	0 (0,0)
63	Main IO 2 Option	Table 8-7 Main IO Option CV Format	3
65,64	Main IO 2 Action	Table 8-8 Main IO Action CV Format	8
67,66	Main IO 2 Secondary	Table 8-9 Main IO Secondary CV Format.	0 (0,0)
68	Aux IO 2 Option	Table 8-10 Aux IO Option CV Format	33

CV #	Description	Range	Default
70,69	Aux IO 2 Action	Table 8-11 Aux IO Action CV Format	4
72,71	Aux IO 2 Secondary	Table 8-12 Aux IO Secondary CV Format.	0 (0,0)
73	Servo 2 Thrown Speed	Table 8-4 Servo Speed CV Format.	4
74-76	Reserved		
78,77	Servo 3 Closed Position	0-2400	1260 (4,236)
80,79	Servo 3 Thrown Position	0-2400	1140 (4,116)
81	Servo 3 Speed	Table 8-4 Servo Speed CV Format.	4
82	Servo 3 Output Option	Table 8-5 Servo Output Option CV Format.	2
84,83	Servo 3 Cascade Address	Table 8-6 Servo Cascade Address CV Format.	0 (0,0)
85	Main IO 3 Option	Table 8-7 Main IO Option CV Format	3
87,86	Main IO 3 Action	Table 8-8 Main IO Action CV Format	32
89,88	Main IO 3 Secondary	Table 8-9 Main IO Secondary CV Format.	0 (0,0)
90	Aux IO 3 Option	Table 8-10 Aux IO Option CV Format	33
92,91	Aux IO 3 Action	Table 8-11 Aux IO Action CV Format	16

CV #	Description	Range	Default
94,93	Aux IO 3 Secondary	Table 8-12 Aux IO Secondary CV Format.	0 (0,0)
95	Servo 3 Thrown Speed	Table 8-4 Servo Speed CV Format.	4
96-98	Reserved		
100,99	Servo 4 Closed Position	0-2400	1260 (4,236)
102,101	Servo 4 Thrown Position	0-2400	1140 (4,116)
103	Servo 4 Speed	Table 8-4 Servo Speed CV Format.	4
104	Servo 4 Output Option	Table 8-5 Servo Output Option CV Format.	2
106,105	Servo 4 Cascade Address	Table 8-6 Servo Cascade Address CV Format.	0 (0,0)
107	Main IO 4 Option	Table 8-7 Main IO Option CV Format	3
109,108	Main IO 4 Action	Table 8-8 Main IO Action CV Format	128
111,110	Main IO 4 Secondary	Table 8-9 Main IO Secondary CV Format.	0 (0,0)
112	Aux IO 4 Option	Table 8-10 Aux IO Option CV Format	33
114,113	Aux IO 4 Action	Table 8-11 Aux IO Action CV Format	64
116,115	AuxIO 4 Secondary	Table 8-12 Aux IO Secondary CV Format.	0 (0,0)

CV #	Description	Range	Default
117	Servo4 Thrown Speed	Table 8-4 Servo Speed CV Format.	4
118-128	Reserved		
130,129	Route 1 Entry 1	Table 8-13 Route CV Format.	0 (0,0)
132,131	Route 1 Entry 2	Table 8-13 Route CV Format.	0 (0,0)
134,133	Route 1 Entry 3	Table 8-13 Route CV Format.	0 (0,0)
136,135	Route 1 Entry 4	Table 8-13 Route CV Format.	0 (0,0)
138,137	Route 1 Entry 5	Table 8-13 Route CV Format.	0 (0,0)
140,139	Route 1 Entry 6	Table 8-13 Route CV Format.	0 (0,0)
142,141	Route 1 Entry 7	Table 8-13 Route CV Format.	0 (0,0)
144,143	Route 1 Entry 8	Table 8-13 Route CV Format.	0 (0,0)
146,145	Route 2 Entry 1	Table 8-13 Route CV Format.	0 (0,0)
148,147	Route 2 Entry 2	Table 8-13 Route CV Format.	0 (0,0)
150,149	Route 2 Entry 3	Table 8-13 Route CV Format.	0 (0,0)
152,151	Route 2 Entry 4	Table 8-13 Route CV Format.	0 (0,0)
154,153	Route 2 Entry 5	Table 8-13 Route CV Format.	0 (0,0)
156,155	Route 2 Entry 6	Table 8-13 Route CV Format.	0 (0,0)

CV #	Description	Range	Default
158,157	Route 2 Entry 7	Table 8-13 Route CV Format.	0 (0,0)
160,159	Route 2 Entry 8	Table 8-13 Route CV Format.	0 (0,0)
162,161	Route 3 Entry 1	Table 8-13 Route CV Format.	0 (0,0)
164,163	Route 3 Entry 2	Table 8-13 Route CV Format.	0 (0,0)
166,165	Route 3 Entry 3	Table 8-13 Route CV Format.	0 (0,0)
168,167	Route 3 Entry 4	Table 8-13 Route CV Format.	0 (0,0)
170,169	Route 3 Entry 5	Table 8-13 Route CV Format.	0 (0,0)
172,171	Route 3 Entry 6	Table 8-13 Route CV Format.	0 (0,0)
174,173	Route 3 Entry 7	Table 8-13 Route CV Format.	0 (0,0)
176,175	Route 3 Entry 8	Table 8-13 Route CV Format.	0 (0,0)
178,177	Route 4 Entry 1	Table 8-13 Route CV Format.	0 (0,0)
180,179	Route 4 Entry 2	Table 8-13 Route CV Format.	0 (0,0)
182,181	Route 4 Entry 3	Table 8-13 Route CV Format.	0 (0,0)
184,183	Route 4 Entry 4	Table 8-13 Route CV Format.	0 (0,0)
186,185	Route 4 Entry 5	Table 8-13 Route CV Format.	0 (0,0)

CV #	Description	Range	Default
188,187	Route 4 Entry 6	Table 8-13 Route CV Format.	0 (0,0)
190,189	Route 4 Entry 7	Table 8-13 Route CV Format.	0 (0,0)
192,191	Route 4 Entry 8	Table 8-13 Route CV Format.	0 (0,0)
193-251	Reserved		
252-255	Reserved JMRI template		

Table 8-1 Quad-LN CV Table.

8.2 Long Address CV Format

CV	17								18							
Bit	1	1	H	h	h	h	h	h	l	l	l	l	l	l	l	l
Default	1	1	1	0	1	0	1	0	1	1	1	1	1	0	0	0

h = High address bits
l = Low address bits

Table 8-2 Long Address CV Format.

8.3 Configuration CV Format

CV	21							
Bit	b	Lb	a	g	l	i	o	s
Default	0	0	1	0	1	1	1	1

b = Busy retry: 0 = Disabled, 1 = Enabled
lb = Lock blink state: 0 = Locked, 1 = Unlocked
a = Address mode: 0 = Short, 1 = Long
g = GPON interrogate: 0 = Off, 1 = On
l = Local Action: 0 = Internal, 1 = Visible on Loconet
i = Input Interrogate: 0 = Off, 1 = On
o = Output Off switch request: 0 = don't send, 1 = send
s = Servo State Memory: 0 = Off, 1 = On

Table 8-3 Configuration CV Format.

8.4 Servo Speed CV Format

CV	37, 59, 81, 103							
Bit	r	r	s	s	s	s	s	s
Default	0	0	0	0	0	1	0	0

r = RapidStart mode: 0 = None, 1 = one-eighth, 2 = one-quarter, 3 = three-eighths
s = Speed: 0 = slowest, 63 = fastest

Table 8-4 Servo Speed CV Format.

8.5 Servo Output Option CV Format

CV	38, 60, 82, 104							
Bit	o	mp	l	d	t	t	m	m
Default	0	0	0	0	0	0	0	0

o = Orientation: 0 = Normal, 1 = Reverse
mp=Midpoint: 0=Disabled, 1 = Enabled
l = Lock mode: 0 = Not Lockable, 1 = Local Lockout
d = Directional Speed: 0=Disabled, 1 = Enabled
t = Trigger: 0 = None, 1 = Closed, 2 = Thrown, 3 = Closed or Thrown
m = Message: 0 = None, 1 = Level Feedback, 2 = Exact Feedback

Table 8-5 Servo Output Option CV Format.

8.6 Servo Cascade Address CV Format

CV	40, 62, 84, 106								39, 61, 83, 105							
Bit	x	x	a	a	x	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x = not used
a = Action: 0 = Closed, 1 = Thrown, 2 = Follow, 3 = Invert
h = High address bits
l = Low address bits

Table 8-6 Servo Cascade Address CV Format.

8.7 Main IO Option CV Format

CV	41, 63, 85, 107							
Bit	f	s	l	l	l	m	t	t
Default	0	0	0	1	0	0	1	0

f = DCC Freeze: 0 = Always live, 1 = Freeze on DCC loss
s = LED Sense: 0 = Normal, 1 = Inverted
l = LED Mode: 0 = Disable, 1 = Steady, 2 = Blink on Move, 3 = Blink on Move or Lock, 4 = Blink on Lock, 5 = Lock State
m = Message: 0 = Sensor, 1 = Closed Position Feedback
t = Trigger: 0 = None, 1 = Positive Edge, 2 = Negative Edge, 3 = Both Edges
x = not used

Table 8-7 Main IO Option CV Format

8.8 Main IO Action CV Format

CV	43, 65, 87, 109								42, 64, 86, 108							
Bit	d	d	d	D	c	c	c	c	b	b	b	b	a	a	a	A
Default	D	D	D	D	C	C	C	C	B	B	B	B	A	A	A	A

d = Servo 4
c = Servo 3
b = Servo 2
a = Servo 1
DDDD, CCCC, BBBB, AAAA: 0 = None, 1 = Toggle, 2 = Follow, 3 = Invert, 4 = Closed, 5 = Thrown, 9 = Lock Toggle, 10 = Lock Follow, 11 = Lock Invert, 12 = Lock Closed, 13 = Lock Thrown
For the Servo in the same group, Default = 2. For all other Servos, Default = 0.

Table 8-8 Main IO Action CV Format

8.9 Main IO Secondary CV Format

CV	45, 67, 89, 111								44, 66, 88, 110							
Bit	t	d	c	C	h	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

t = Type: 0 = Trigger on this message, 1 = Send this message
d = Device: 0 = Turnout, 1 = Sensor
c = Condition: 0 = None, 1 = Hi/Closed, 2 = Lo/Thrown, 3 = Both/Follow
h = high address bits
l = low address bits

Table 8-9 Main IO Secondary CV Format.

8.10 Aux IO Option CV Format

CV	46, 68, 90, 112							
Bit	f	s	l	l	l	m	t	t
Default	0	0	0	0	0	0	1	1

f = DCC Freeze: 0 = Always live, 1 = Freeze on DCC loss
s = LED Sense: 0 = Normal, 1 = Inverted
l = LED Mode: 0 = Disable, 1 = Steady, 2 = Blink on Move, 3 = Blink on Move or Lock, 4 = Blink on Lock, 5 = Lock State
m = Message: 0 = Sensor, 1 = Thrown Position Feedback
t = Trigger: 0 = None, 1 = Positive Edge, 2 = Negative Edge, 3 = Both Edges

Table 8-10 Aux IO Option CV Format

8.11 Aux IO Action CV Format

CV	48, 70, 92, 114								47, 69, 91, 113							
Bit	d	d	d	d	c	c	c	c	b	b	b	b	a	a	a	A
Default	D	D	D	D	C	C	C	C	B	B	B	B	A	A	A	A

d = Servo 4
c = Servo 3
b = Servo 2
a = Servo 1
DDDD, CCCC, BBBB, AAAA: 0 = None, 1 = Toggle, 2 = Follow, 3 = Invert, 4 = Closed, 5 = Thrown, 9 = Lock Toggle, 10 = Lock Follow, 11 = Lock Invert, 12 = Lock Closed, 13 = Lock Thrown
For the Servo in the same group, Default = 1. For all other Servos, Default = 0.

Table 8-11 Aux IO Action CV Format

8.12 Aux IO Secondary CV Format

CV	50, 72, 94, 116								49, 71, 93, 115							
Bit	t	d	c	c	h	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

t = Type: 0 = Trigger on this message, 1 = Send this message
d = Device: 0 = Turnout, 1 = Sensor
c = Condition: 0 = None, 1 = Hi/Closed, 2 = Lo/Thrown, 3 = Both/Follow
h = high address bits
l = low address bits

Table 8-12 Aux IO Secondary CV Format.

8.13 Route CV Format

CV	130-192 (even)								129-191 (odd)							
Bit	t	t	s	s	h	h	h	h	l	l	l	l	l	l	l	l
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

t = Type: 0 = None, 1 = Basic, 2 = Selector, 3 = Expand
s = Sense/Device: 0 = Closed Turnout, 1 = Thrown Turnout, 2 = Hi Sensor, 3 = Lo Sensor
c = Condition: 0 = None, 1 = Hi/Closed, 2 = Lo/Thrown, 3 = Both/Follow
h = high address bits
l = low address bits

Type applies to Entry 1 only, otherwise not used. Type = 3 is not used on Route 1.

Table 8-13 Route CV Format.

9 JMRI and Turnout Feedback

The discussion is this section is only intended for technical users trying to precisely understand and control turnout icon behavior in JMRI. Most users will get acceptable results with JMRI panels by setting the Quad-LN to provide simulated Exact Feedback (Section 3.4.1), setting the turnout feedback type in JMRI to Exact, and then selecting the tristate option on any turnout icon that you want to show a special indication when the turnout is in motion. If this approach works for you, then ignore the reset of this section.

Turnouts in JMRI have a Commanded state and a Known (Feedback) state. JMRI gathers information about the state of a turnout by monitoring loconet messages. The interpretation of certain messages varies based on the turnout feedback type you select. There are subtle nuances to the behavior, for example the Commanded state might not reflect the last Switch command that was sent.

9.1 Switch Request

The command that initiates a turnout position change is the Loconet SWITCH REQUEST (OPC_SW_REQ). OPC_SW_REQ messages have an Output ON version and an Output OFF version. The standard implementation of a Switch command actually sends two OPC_SW_REQ messages: the first with Output ON and the second about 100 milliseconds later with Output OFF.

9.2 Turnout feedback

Turnout position feedback reporting is provided via the Loconet Turnout SENSOR state REPORT (OPC_SW_REP). There are two different forms of this report: OUTPUT levels feedback and INPUT levels feedback. The OUTPUT form was intended to provide the state of the drive to the turnout motor while the INPUT form was meant to provide the state of sensors monitoring the physical position of the points. Since both forms can report one of three (really four) states, either can be used with the Quad-LN. JMRI responds a bit differently to each type of feedback, however, so it is worthwhile to characterize the two approaches and consider their differences. The following data was collected using JMRI 2.14.

9.2.1 Approach 1. OUTPUT form of OPC_SW_REP

To use the OUTPUT form of OPC_SW_REP, the JMRI feedback mode is set to Monitoring. In Monitoring mode, both the Commanded State and the Feedback State follow the OPC_SW_REQ and OUTPUT form OPC_SW_REP messages seen on Loconet. Here is a typical sequence assuming the turnout starts as Thrown:

Num	Sender	Loconet message	Commanded/Feedback state
0			Thrown
1	JMRI	OPC_SW_REQ Closed Output ON	Closed
2	JMRI	OPC_SW_REQ Closed Output OFF	Closed
3	Quad	OPC_SW_REP Closed=Off, Thrown=Off	Unknown (in transit)
4	Quad	OPC_SW_REP Closed=On, Thrown=Off	Closed

5	JMRI	OPC_SW_REQ Thrown Output ON	Thrown
6	JMRI	OPC_SW_REQ Thrown Output OFF	Thrown
7	Quad	OPC_SW_REP Closed=Off, Thrown=Off	Unknown (in transit)
8	Quad	OPC_SW_REP Closed=Off, Thrown=On	Thrown

Messages 1-2 are sent when you initiate the turnout change. Message 3 is sent by the Quad to indicate that the servo is in motion. Message 4 is sent when the servo finally reaches the endpoint. Messages 5-8 illustrate the transition from Closed to Thrown.

In Monitoring mode the Commanded State tracks the Feedback State. Both states immediately take on the requested (future) position **immediately upon receipt** of the Switch command. Also **both states become Unknown** when the feedback indicates the turnout is in movement. When viewing the Turnout Table, it might be confusing to see the Commanded State shown as Unknown.

Panel icons in JMRI tied to the Loconet Turnout display the Known (Feedback) State as shown in the table. If the tristate option is selected, the icon will show the Unknown state where indicated. Otherwise the icon ignores the Unknown state and as a result **immediately shows the requested** (future) state upon Message 1.

Messages 1-3 typically happen quickly, however the erroneous turnout state change to Closed in 1 and then the change to the proper Unknown state in message 3 can be seen in JMRI when viewing the Turnout table or panel icons.

9.2.2 Approach 2. INPUT form of OPC_SW_REP

To use the INPUT form of OPC_SW_REP, the JMRI feedback mode is set to Exact. In Exact mode, the Commanded State tracks the OPC_SW_REQ messages or follows changes in the Feedback State when OPC_SW_REQ messages are absent. The Feedback State tracks INPUT OPC_SW_REP messages. Here is a typical sequence assuming the turnout starts as Thrown:

Num	Sender	Loconet message	Commanded	Feedback	Icon (tristate)
0			Thrown	Thrown	Thrown
1	JMRI	OPC_SW_REQ Closed Output ON	Closed	Thrown	Unknown
2	JMRI	OPC_SW_REQ Closed Output OFF	Closed	Thrown	Unknown
3	Quad	OPC_SW_REP Aux=Off	Closed	Thrown	Unknown
4	Quad	OPC_SW_REP Switch=On	Closed	Closed	Closed
5	JMRI	OPC_SW_REQ Thrown Output ON	Thrown	Closed	Unknown
6	JMRI	OPC_SW_REQ Thrown Output OFF	Thrown	Closed	Unknown
7	Quad	OPC_SW_REP Switch=Off	Thrown	Closed	Unknown
8	Quad	OPC_SW_REP Aux=On	Thrown	Thrown	Thrown

Messages 1-2 are sent when you initiate the turnout change. 3 is sent by the Quad to indicate that the servo is in motion. 4 is sent when the servo finally reaches the endpoint. Messages 5-8 illustrate the transition from Closed to Thrown.

The JMRI 2.14 implementation of Exact mode only acts upon the INPUT form OPC_SW_REP messages that indicate a sensor change from Off to On. These messages occur at the completion of movement. In practice this means that the Feedback State retains the previous value until the **end** of the movement. The Commanded State changes with the OPC_SW_REQ, so it takes on the new value at the **start** of the movement. Although the Commanded State and Feedback state differ during movement, **neither state is ever Unknown**. As a result, the states shown in the Turnout Table always make sense in this approach.

Panel icons in JMRI tied to the Loconet Turnout display the Known (Feedback) State. If the tristate option is selected, the icon will show Unknown when the Feedback State does not equal the Commanded State (i.e. during movement). If the tristate option is not selected the icon ignores this condition and **continues to show the previous state** until the movement is reported as completed.

9.2.3 Comments

Approach 2 with Exact mode appears to provide the cleanest state feedback. One icon tied to the Loconet Turnout and set for tristate shows the Feedback state in both approaches. A second icon is required if it is desired to show the Commanded State during movement. This is easiest to do with Approach 1 where the second icon can also be tied to the Loconet Turnout but not set to tristate. For Approach 2 the second icon must be tied to an Internal Turnout that reflects the commanded state and drives the loconet turnout via a logix. This Internal Turnout typically already exists when implementing a CTC panel.